




Sie sind hier: **Home**



JSF hat leider die Eigenart, dass der `<h:form>`-Tag ein hidden input-Feld mit dem Attribut "autocomplete" generiert, der ein ungültiges Attribut bei XHTML ist:

```
<input type="hidden" name="javax.faces.ViewState"
id="javax.faces.ViewState" value="-4427180260951230162:-
2730491338021367514" autocomplete="off" />
```

Beim W3C-Validator sieht das ganze dann ungefähr so aus:

 **Line 35, Column 256: there is no attribute "autocomplete"**

```
...wState" value="-4427180260951230162:-2730491338021367514" autocomplete="off" />
```

Man kann dieses Verhalten aber leicht unterbinden. Dazu muss man die Datei web.xml um 4 Zeilen ergänzen:

```
1 <context-param>
2   <param-name>com.sun.faces.autoCompleteOffOnViewState</param-name>
3   <param-value>>false</param-value>
4 </context-param>
```



Tags: [jsf](#), [tipp](#), [Web](#), [xhtml](#)

[keine
Kommentare](#)



Eine moderne Website muss nicht nur auf allen aktuellen Browsern lauffähig sein, sondern soll auch auf Smartphones laufen. Eine Website für ein Handy anzupassen stellt jedoch besondere Herausforderungen an den Entwickler. Einerseits gibt es wieder verschiedene Browser die HTML und CSS auf mehr oder weniger standardkonforme Weise interpretieren. Andererseits gibt es ein paar Besonderheiten, auf die man beim Entwickeln achten muss - beispielsweise:

- Bei mobilen Datenverbindungen wie UMTS oder GPRS ist der Overhead für einen zusätzlichen Request wesentlich höher als bei "normalen"

Verbindungen wie DSL (d. h. so wenige Requests wie möglich).

- Die Downloadgeschwindigkeit mobiler Verbindungen ist geringer (d. h. so wenig Datenmenge wie möglich).
- Mobile Geräte haben eine langsamere CPU (d. h. rechenintensive Elemente, wie table vermeiden).

Herauszufinden, ob eine Seite für mobile Geräte optimiert ist, ist aber aufgrund der Vielzahl an Geräten, Verbindungstypen, etc. schwierig. Es gibt aber eine Seite, die einem bei der Optimierung hilft: [mobiReady \(www.ready.mobi\)](http://www.ready.mobi).

Der Seiten Test gibt Auskunft wie viele Daten für eine Abruf der Seite nötig sind, prüft den Quellcode auf Standardkompatibilität und Kompatibilität zum [XHTML Mobile Profil](#) und gibt Tipps wie man die Seite auf mobilen Geräten beschleunigen kann.



Tags: [css](#), [iPhone](#), [tipp](#), [Web](#)

[keine
Kommentare](#)



Wer unter Windows 7 den Startmenü Ordner im Ordner des Benutzers sucht wird enttäuscht. Er befindet sich nun unter:



%SYSTEMDRIVE%\Users\[Benutzer]\AppData\Roaming\Microsoft\Windows\Start Menu

Dort findet sich ebenfalls der "Senden An" (SendTo) Ordner.

Das Startmenü für alle Benutzer befindet sich auch an einem neuen Ort:

%SYSTEMDRIVE%\ProgramData\Microsoft\Windows\Start Menu

Generell finden sich nun alle benutzerunabhängigen Programmdaten (z.B. Einstellungen) unter %SYSTEMDRIVE%\ProgramData\. Der Ordner %SYSTEMDRIVE%\Users\All Users dient nur mehr der Rückwärtskompatibilität.

Die Benutzer-Ordner der Systemaccounts (z.B. NetworkService) finden sich unter:
%SYSTEMDRIVE%\Windows\ServiceProfiles\



Tags: [microsoft](#), [tipp](#), [windows 7](#)

[keine
Kommentare](#)



In einem der letzten Artikel habe ich gezeigt, wie



man einen eigenen Firefox Sync Server einrichten kann. Wer diesen Server per SSL absichern möchte, wird vor ein paar Komplikationen gestellt. Das äußert sich darin, dass das Sync Plugin regelmäßig meldet, dass Benutzername und oder Passwort falsch sind bzw. Firefox Home auf dem iPhone einen Kommunikationsfehler anzeigt. Dies kann man jedoch leicht beheben, indem man dafür sorgt, dass der Sync Server sowohl mit, als auch ohne SSL erreichbar ist (Wenn man mod_wsgi mit Apache2

einsetzt und die Konfiguration kopiert, darf man nicht vergessen der WSGIProcessGroup einen anderen Namen zu geben, sonst verweigert der Apache die Arbeit).

Verwendet man ein selbst signiertes SSL-Zertifikat, muss man dieses im Firefox als dauerhafte Ausnahme speichern, damit es mit der Synchronisation klappt. Am iPhone gilt Ähnliches: Man muss das selbst signierte Zertifikat als vertrauenswürdiges Zertifikat im iPhone installieren. Dazu kann man beispielsweise das Zertifikat im Firefox exportieren und es sich per Mail senden. Wenn man den Anhang öffnet, wird man vom iPhone direkt gefragt, ob man das Zertifikat installieren möchte. Nach dem Klick auf "installieren" ist das Zertifikat importiert und Firefox Home verrichtet wie gewünscht seinen Dienst.



Tags: [Cloud](#), [firefox](#), [server](#), [Web](#)

[keine
Kommentare](#)



Alle die Hilfe bei der Installation und Konfiguration werden ab jetzt in den [Zarafa Mail Checker FAQ](#) fündig. Sie sind zwar noch nicht allzu ausführlich, dies wird sich jedoch im Laufe der Zeit ändern.

Jeder Probleme mit dem Zarafa Mail Checker hat kann auch gerne eine E-Mail an [office\(at\)sceed.at](mailto:office(at)sceed.at) schreiben.



Tags: [firefox](#), [Web](#), [zarafa](#)

[keine
Kommentare](#)



Firefox bietet seit Version 4 die Möglichkeit die Surf-Chronik, die Passwörter, geöffneten Tabs und die Lesezeichen über verschiedene Geräte hinweg zu synchronisieren. Prinzipiell stellt Mozilla die notwendige Server-Infrastruktur zur Verfügung um die Synchronisierung zu ermöglichen. Wer jedoch, z.B. die Daten lieber bei sich behalten möchte kann auch seinen eigenen Sync-Server einrichten.

Diese Anleitung basiert auf Debian 6.0.3 (squeeze) zusätzlich werden ein eingerichteter Apache2-Server und ein MySQL-Server vorausgesetzt.

Zusätzlich benötigt man folgende Pakete:

- python-dev
- make
- mercurial
- python-mysqldb
- libapache2-mod-wsgi

Diese muss man zunächst einmal installieren:

```
1 apt-get install python-dev mercurial sqlite3 python-virtualenv python-mysqldb
2 /usr/bin/easy_install Mysql-Python
```

Zunächst muss man sich die letzte Version herunterladen und

```
1 mkdir /opt
2 cd /opt
3 hg clone https://hg.mozilla.org/services/server-full
4 cd server-full
5 make build
```

Der Server ist nun grundsätzlich installiert und lauffähig.

Mittels `bin/paster serve development.ini` kann man den Server (der in der Standard-Konfiguration SQLite verwendet) starten. Er sollte anschließend unter `http://127.0.0.1:5000` erreichbar sein und prinzipiell funktionieren. Der inkludierte Webserver sollte jedoch nicht im Produktiveinsatz verwendet werden, da die Performance nicht optimal ist.

Zunächst muss man eine MySQL-Datenbank und einen Benutzer der Zugriff auf die Datenbank hat erstellen. Nun muss man die Konfiguration anpassen. Das Config-File findet sich unter `/opt/server-full/etc/sync.conf`.

```
1 [storage]
2 backend = syncstorage.storage.sql.SQLStorage
3 sqluri = mysql://[User]:[Passwort]@localhost:3306/[Datenbank]
4 standard_collections = false
5 use_quota = true
6 quota_size = 5120
7 pool_size = 100
8 pool_recycle = 3600
9 reset_on_return = true
10 display_config = true
11 create_tables = true
12
13 [auth]
14 backend = services.auth.sql.SQLAuth
15 sqluri = mysql://[User]:[Passwort]@localhost:3306/[Datenbank]
16 pool_size = 100
17 pool_recycle = 3600
18 create_tables = true
19 fallback_node = <a href="http://www.url-zum-server.tld/sync">http://www.url-z
```

Die Parameter `backend` bestimmen, auf welche Art die Daten gespeichert bzw. die Authentifizierung durchgeführt wird. `sqluri` bestimmt Datenbank, Server und Benutzerdaten für den Datenbankserver. Über `user_quota` bzw. `quota_size` kann ein Speicherlimit für die Nutzer festgelegt werden. Mit `pool_size` und `pool_recycle` kann das SQL-Connection-Pooling konfiguriert werden. `create_tables` sorgt dafür, dass die

Datenbanktabellen erstellt werden, wenn sie noch nicht existieren. `fallback_node` erwartet die öffentliche URL zum Sync-Server.

Zusätzlich hat man noch die Möglichkeit reCAPTCHA zu aktivieren, um automatische Anmeldungen zu verhindern:

```
1 [captcha]
2 use = true
3 public_key = [Public_Key]
4 private_key = [Private_Key]
5 use_ssl = true
```

Public- und Private-Key erhält man auf der Website von [reCAPTCHA](#).



Zunächst werden ein Benutzer und eine Gruppe benötigt, die den Server-Prozess ausführen:

```
1 groupadd ffsync
2 useradd -g ffsync -d /opt/server-full -s /bin/false ffsync
```

Der neue User braucht auch die Rechte, um auf den Server zugreifen zu dürfen:

```
1 chown ffsync:ffsync /opt/server-full -R
2 chmod 400 /opt/server-full/etc/sync.conf
```

Jetzt muss man die Apache-Module Alias und WSGI aktivieren:

```
1 a2enmod alias wsgi
```

Nun noch den virtual host konfigurieren:

```
1 WSGIDaemonProcess sync user=ffsync group=ffsync processes=2 threads=2
2 WSGIScriptAlias /sync /opt/server-full/sync.wsgi
3 WSGIPassAuthorization On
4 WSGIProcessGroup sync
```



Zuletzt muss man noch den Apache-Server neu starten:

```
/etc/init.d/apache2 restart
```

Jetzt sollte der Sync Server einsatzbereit sein.



- [Englisches Tutorial zum Einrichten des Sync Servers von Mozilla - http://docs.services.mozilla.com/howtos/run-sync.html](http://docs.services.mozilla.com/howtos/run-sync.html)
- [Konfigurations-Dokumentation des Sync Servers - http://docs.services.mozilla.com/server-devguide/configuration.html#configuration](http://docs.services.mozilla.com/server-devguide/configuration.html#configuration)
- [mod_wsgi Dokumentation - https://code.google.com/p/modwsgi/wiki/WhereToGetHelp?tm=6](https://code.google.com/p/modwsgi/wiki/WhereToGetHelp?tm=6)



Tags: [Cloud](#), [firefox](#), [server](#), [Web](#)

[keine
Kommentare](#)



Ich denke viel kann man zu dem Titel nicht mehr sagen.
Für alle die noch in Weihnachtsstimmung kommen wollen:



veröffentlicht unter: [Allgemein](#)

[keine
Kommentare](#)



Vermutlich als Weihnachtsgeschenk hat Mozilla pünktlich Firefox 9 veröffentlicht. Passend dazu gibt es nun auch eine angepasste Version des Zараfa Mail Checkers.

Neben der Anpassung an die neue Firefox Version gibt es diesmal auch eine kleine Neuerung: Über das Kontextmenü in der Add-on-Leiste kann man nun direkt eine neue Mail verschicken.

Alles Weitere und den Download findet Ihr auf der [Zараfa Mail Checker Projektseite](#).

Viel Spaß damit!



Tags: [firefox](#), [Web](#), [zarafa](#)

[keine
Kommentare](#)



Kontaktformulare, Kommentare und wann auch immer man auf öffentlichen Seiten Userinput verarbeitet sollte man seine Formulare mit einem CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) vor SPAM schützen. Wie Forscher der Stanford University herausfanden sind diese nicht unbedingt mehr gegen automatische Attacken gefeit. Ganz im Gegenteil es gelang ihnen 13 von 15 bekannten Systeme auszutricksen ([mehr dazu](#)). Lediglich Googles Captcha und reCAPTCHA widerstanden den



Angriffsversuchen.
Wer nun eine einfache Möglichkeit sucht reCAPTCHA in seine JSF-Website einzubinden ist bei der PrimeFaces-Bibliothek besonders gut aufgehoben.



Zunächst muss man sich bei reCAPTCHA registrieren und seine Seite anlegen:

<http://www.google.com/recaptcha>

Anschließend muss man PrimeFaces in sein Projekt integrieren. Hier kann man die Bibliothek herunterladen bzw. findet man das Maven-Repository:

<http://www.primefaces.org/downloads.html>

Anschließend muss der Namespace in das Faclet eingebunden werden, in dem das Captcha angezeigt werden soll:

```
1 <ui:composition xmlns="http://www.w3.org/1999/xhtml"
2   xmlns:ui="http://java.sun.com/jsf/facelets"
3   xmlns:pp="http://primefaces.prime.com.tr/ui">
4   ...
```

Anschließend bettet man das Control in den Code ein:

```
1 <pp:tcaptcha label="Sicherheitsprüfung" language="de" theme="white"
2   secure="true" required="true" requiredMessage="#{msg.CaptchaRequired}"/>
```

Das Attribut `required` sorgt dafür, dass das Captcha zwingend angegeben werden muss. Mit `requiredMessage` wird die entsprechende Fehlermeldung definiert.

`secure` aktiviert die SSL-Unterstützung.

Über das `language`-Attribut kann die Anzeigesprache festgelegt werden. Folgende sind verfügbar:

- `de` - Deutsch
- `en` - Englisch
- `es` - Spanisch
- `fr` - Französisch
- `nl` - Niederländisch
- `pt` - Portugiesisch
- `ru` - Russisch
- `tr` - Türkisch

Zuletzt kann über das `theme`-Attribut das Aussehen des Captchas spezifiziert werden. Dabei hat man folgende Möglichkeiten:

reCAPTCHA - Theme: red (standard)

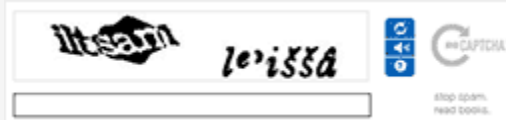


reCAPTCHA - Theme: white



reCAPTCHA - Theme: blackglass

reCAPTCHA - Theme: clean



reCAPTCHA - public & private Key

Zuletzt müssen der Private- und der Public-Key den man bei der Anmeldung erhalten hat in der web.xml des Projekts hinterlegt werden:

```
1 <context-param>
2   <param-name>primefaces.PUBLIC_CAPTCHA_KEY</param-name>
3   <param-value>[PUBLIC-Key]</param-value>
4 </context-param>
5 <context-param>
6   <param-name>primefaces.PRIVATE_CAPTCHA_KEY</param-name>
7   <param-value>[PRIVATE-Key]</param-value>
8 </context-param>
```

Fehlermeldung spezifizieren

Zuletzt kann man noch die Meldung anpassen die der Benutzer zu sehen bekommt wenn er sich vertippt. Dafür muss man zunächst in der faces-config.xml ein Message-Bundle spezifiziert werden, mit dem man die Standardmeldung der Validatoren überschreiben kann:

```
1 <?xml version='1.0' encoding='UTF-8'?>
2 <faces-config version="2.0"
3   xmlns="http://java.sun.com/xml/ns/javaee"
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5   xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com
6
7   <application>
8     <message-bundle>ValidationMessages</message-bundle>
9
10    ...
11
12  </application>
13 </faces-config>
```

Im Message-Bundle kann man nun die eigentliche Fehlermeldung spezifizieren:

```
1 ...
2 primefaces.captcha.INVALID=Die Sicherheitsfrage wurde falsch beantwortet.
3 primefaces.captcha.INVALID_detail=Die Sicherheitsfrage wurde falsch beantw
4 ...
```



- PrimeFaces Komponenten-Bibliothek - <http://www.primefaces.org/>
- reCAPTCHA - www.recaptcha.net
- reCAPTCHA Dokumentation - <https://code.google.com/intl/de-DE/apis/recaptcha/intro.html>



Tags: [jsf](#), [Sicherheit](#), [tipp](#), [Web](#)

[keine
Kommentare](#)



Ab April 2012 tritt auch in Österreich die Vorratsdatenspeicherung in Kraft. Das bedeutet, dass ab diesem Tag von jedem Bürger erfasst wird z.B. von wo er wie lange mit wem telefoniert hat, oder von jeder Mail Zeitpunkt des Sendens, Absender und Empfänger protokolliert werden. Diese Aufzeichnung finden unabhängig von konkreten Verdachtsmomenten statt - es werden also die Daten von wirklich jedem erfasst. Diese Daten bleiben anschließend sechs Monate gespeichert.

IMHO wird durch diese verdachtsunabhängige Speicherung eigentlich jeder unter Generalverdacht gestellt. Zusätzlich ist es für mich fraglich ob damit überhaupt Kriminelle sinnvoll ermittelt werden können, denn jedem auch nur minimal

versierten Nutzer werden vermutlich sofort unzählige Möglichkeiten einfallen die Vorratsdatenspeicherung zu umgehen (z.B. offene WLANs). Das heißt, das Gros der Kriminellen wird vermutlich niemals erfasst.



Der AKVorrat hat eine Bürgerinitiative ins Leben gerufen, um die Vorratsdatenspeicherung noch abwenden zu können. Jeder der die Vorratsdatenspeicherung noch aufhalten möchte kann unter <https://zeichnemit.at/> die Bürgerinitiative unterstützen.



- AKVorrat - <http://www.akvorrat.at/>
- EU-Richtlinie 2006/24/EG zur Vorratsdatenspeicherung - <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32006L0024:de:HTML>
- Heise Berichterstattung zur Vorratsdatenspeicherung - <http://www.heise.de/newsticker/meldung/Oesterreich-Unterschrift-gegen-Vorratsdatenspeicherung-online-moeglich-1399245.html>



Tags: [Datenschutz](#), [Vorratsdatenspeicherung](#)

[keine
Kommentare](#)

» **l e k i t r A e r e t l ä**

Seiten

[Zarafa Mail Checker](#)

[Zarafa Mail Checker FAQ](#)

[Tools, Infos und Links](#)

[IMHO: Stoppt den Internet Explorer 6](#)

[Impressum](#)

Kategorien

[Allgemein](#)

[Apple](#)

[Datenschutz](#)

[Linux](#)

[MySQL](#)

[Tipps](#)

[Tutorials](#)

[Web](#)

[JSF](#)

[WordPress](#)

[Zarafa Mail Checker](#)

Blogroll

[sceed](#)

Archiv

[Januar 2012](#)

[Dezember 2011](#)

[November 2011](#)

[Oktober 2011](#)

Meta

[Anmelden](#)

[RSS](#)

[RSS \(Kommentare\)](#)